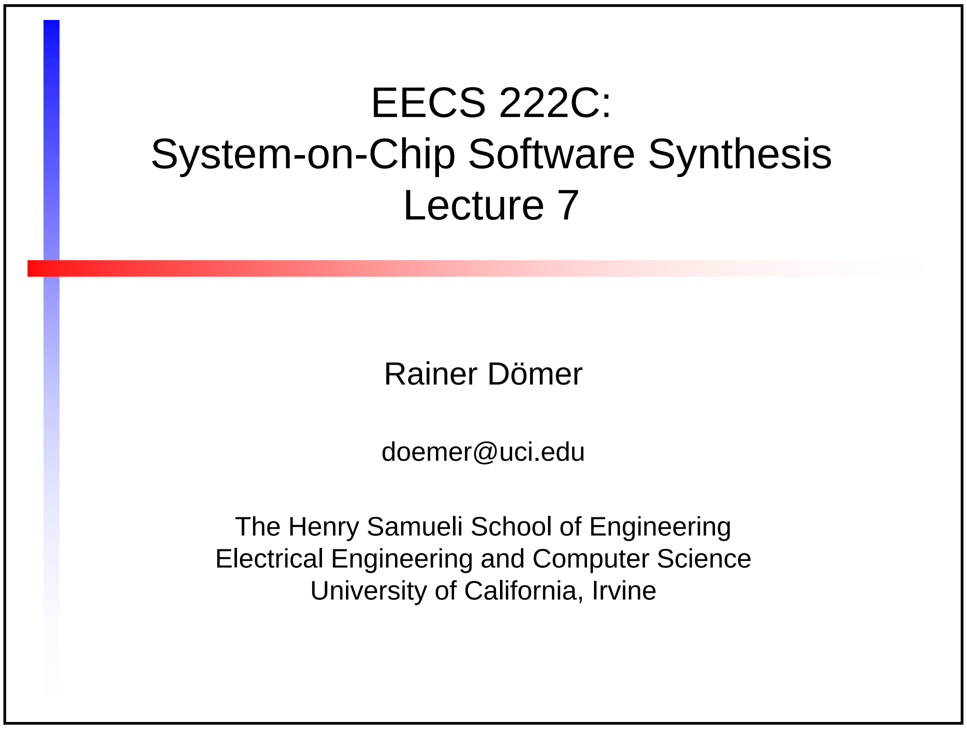
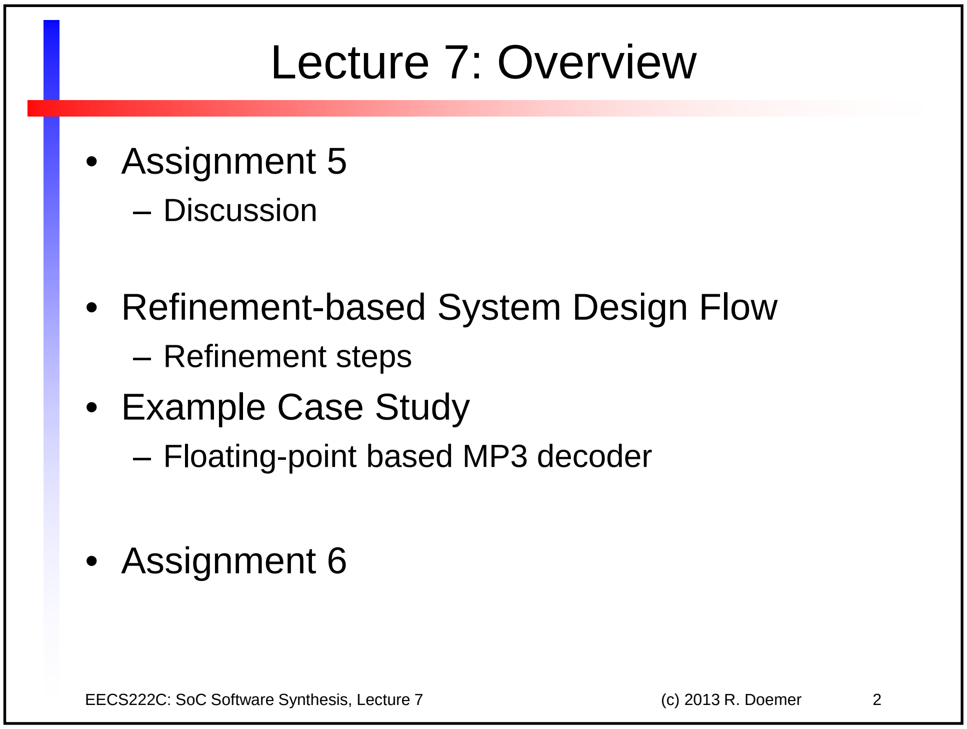


# EECS 222C:
# System-on-Chip Software Synthesis
# Lecture 7

Rainer Dömer

doemer@uci.edu

The Henry Samueli School of Engineering
Electrical Engineering and Computer Science
University of California, Irvine



## Lecture 7: Overview

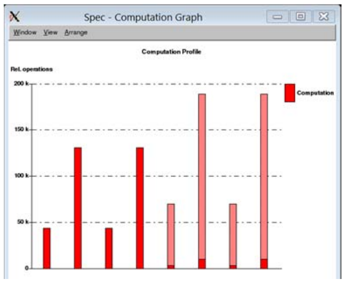
- Assignment 5
  - Discussion
- Refinement-based System Design Flow
  - Refinement steps
- Example Case Study
  - Floating-point based MP3 decoder
- Assignment 6

# Assignment 5

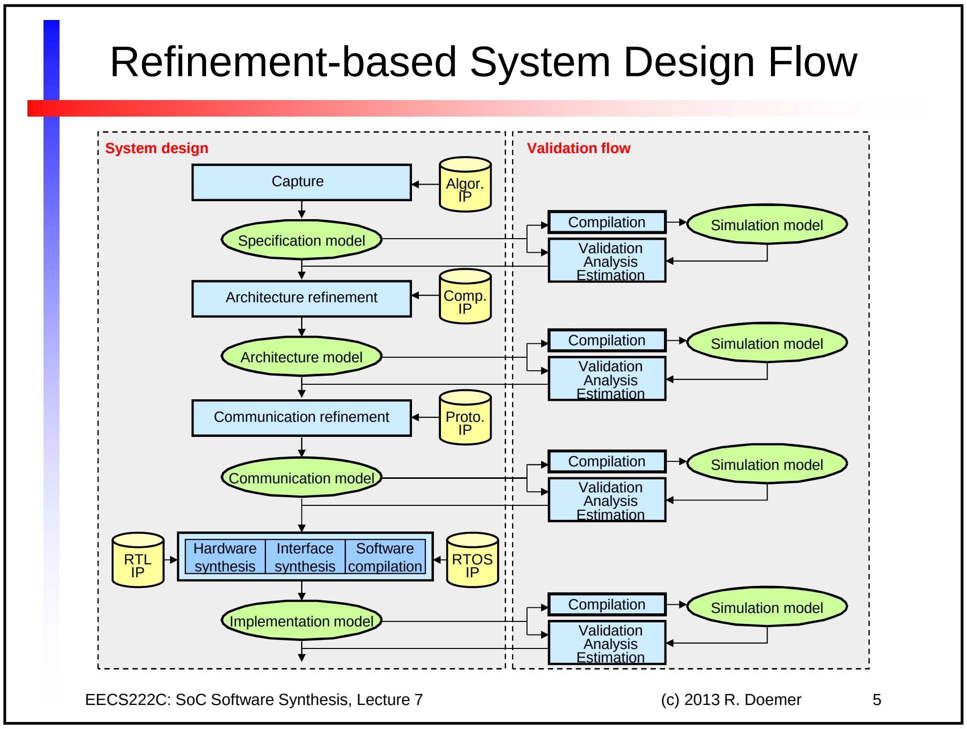
1. Profile your MP3 Decoder model in SCE
   - (continued from previous assignment)
   - Load your MP3 project in SCE
     - ➢ **`Project->Load “mp3.sce”`**
   - Open your “Spec” design model and validate it
     - ➢ Double-click on **`Spec.sir`** in the project window
     - ➢ **`Validation->Compile`**
     - ➢ **`Validation->Simulate`**
   - Profile your MP3 decoder in SCE
     - ➢ **`Validation->Profile`**

# Assignment 5

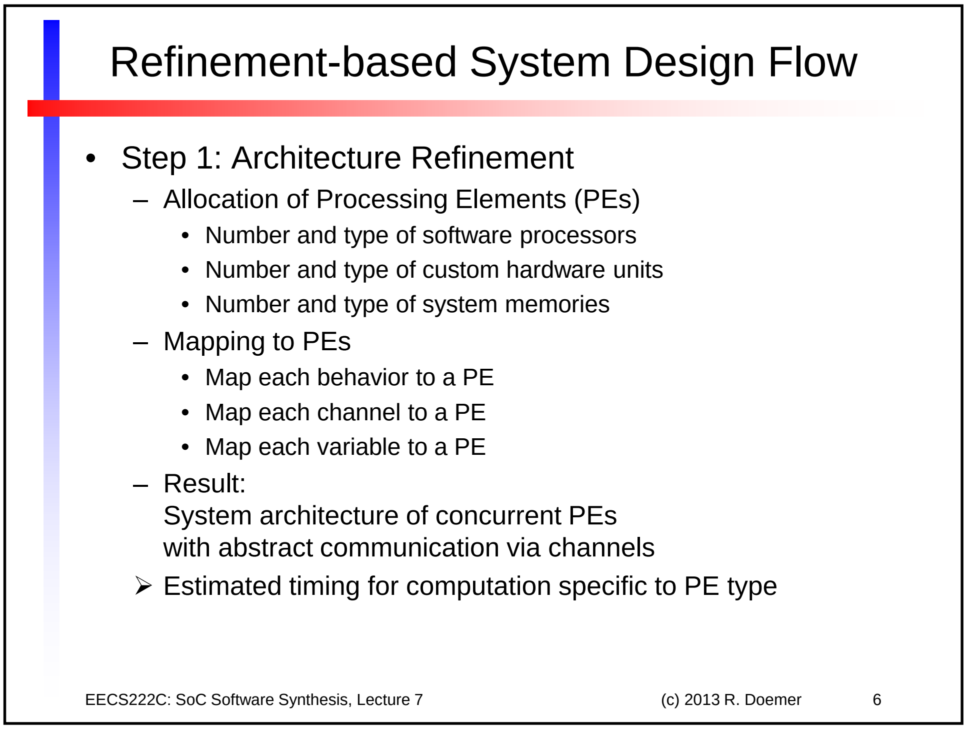
2. Analyze your Profiling Results
   - Use the SCE bar charts to compare the computational complexity of the behaviors in your MP3 decoder model
     - ➢ In the hierarchy browser, select behaviors of interest (use CTRL-LeftClick to select/deselect)
     - ➢ **`RightClick->Graphs->Computation`**
   - Identify the behavior instances with the most computational load
     - Goal is to find those components that make good candidates for hardware accelleration
       - Short code
       - Regular structure
       - High computation
     - Hint: There are 8 candidates as shown in the chart on the right!
   - Deliverable
     - **`ComputationProfile.pdf`**



Example Computation Profile
(block names omitted)

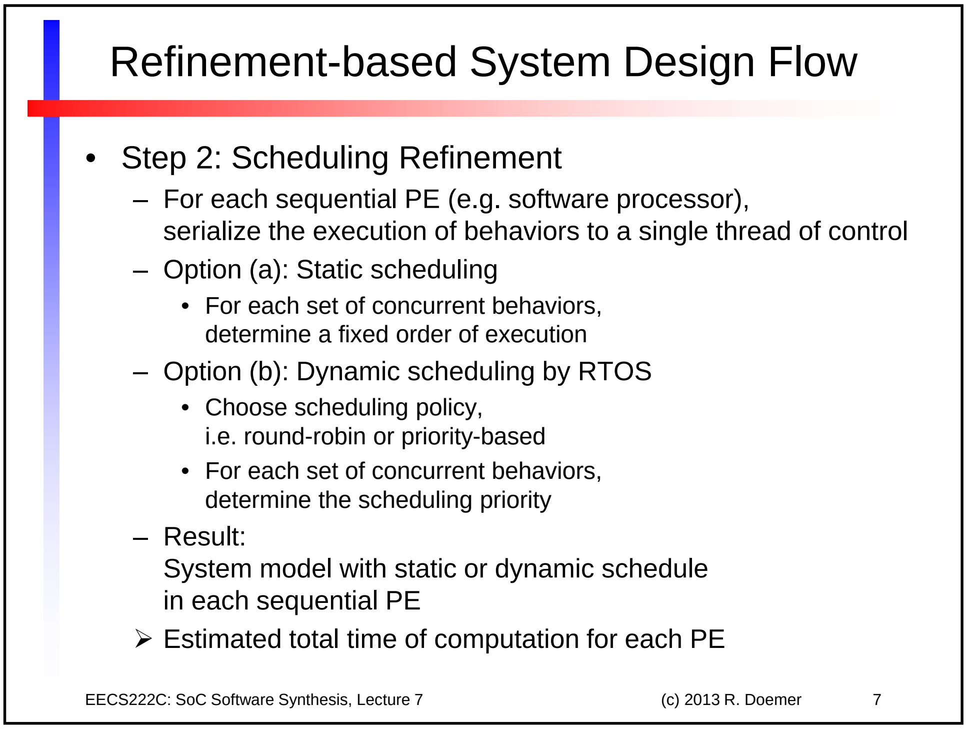


## Refinement-based System Design Flow



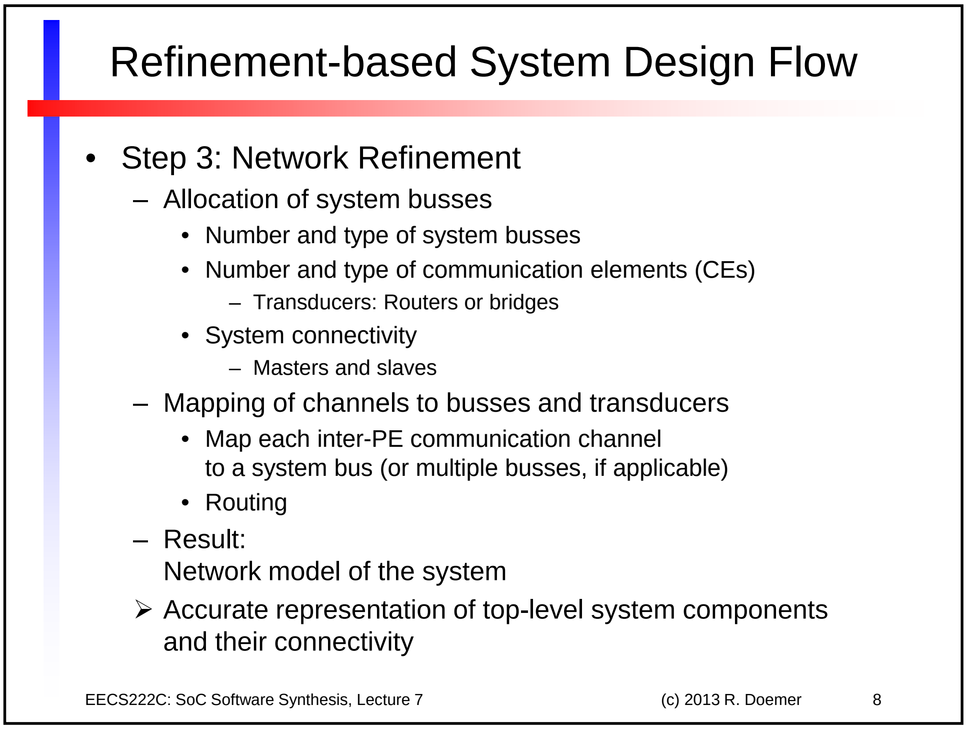
## Refinement-based System Design Flow

- Step 1: Architecture Refinement
  - Allocation of Processing Elements (PEs)
    - Number and type of software processors
    - Number and type of custom hardware units
    - Number and type of system memories
  - Mapping to PEs
    - Map each behavior to a PE
    - Map each channel to a PE
    - Map each variable to a PE
  - Result:
    System architecture of concurrent PEs
    with abstract communication via channels
  - ➢ Estimated timing for computation specific to PE type



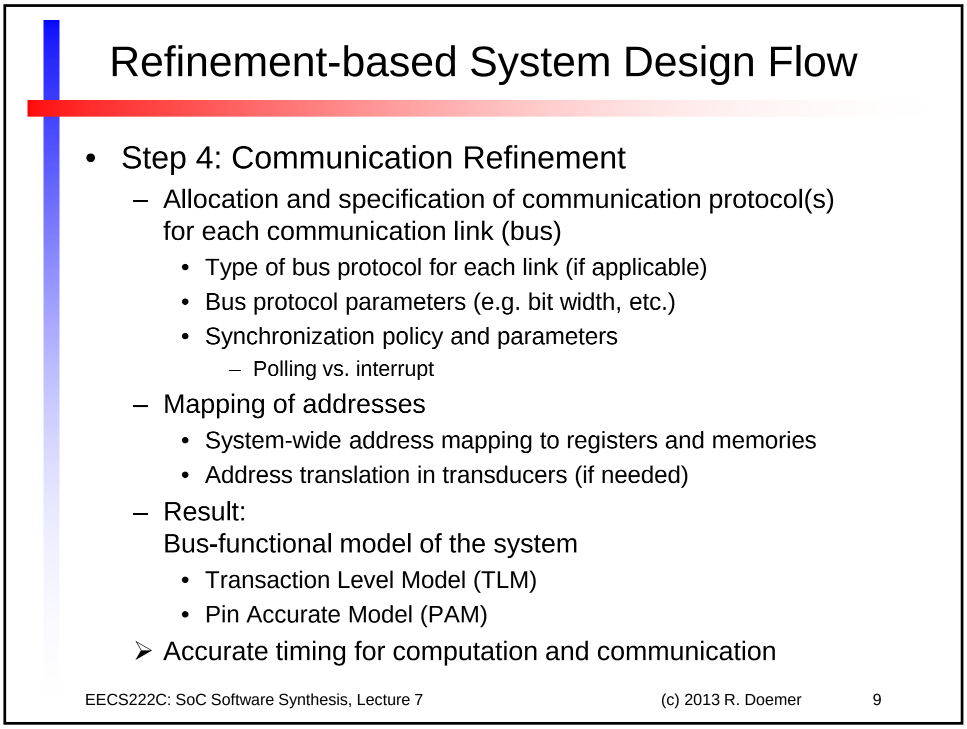
## Refinement-based System Design Flow

- Step 2: Scheduling Refinement
  - For each sequential PE (e.g. software processor), serialize the execution of behaviors to a single thread of control
  - Option (a): Static scheduling
    - For each set of concurrent behaviors, determine a fixed order of execution
  - Option (b): Dynamic scheduling by RTOS
    - Choose scheduling policy, i.e. round-robin or priority-based
    - For each set of concurrent behaviors, determine the scheduling priority
  - Result:
    System model with static or dynamic schedule in each sequential PE
  - ➢ Estimated total time of computation for each PE



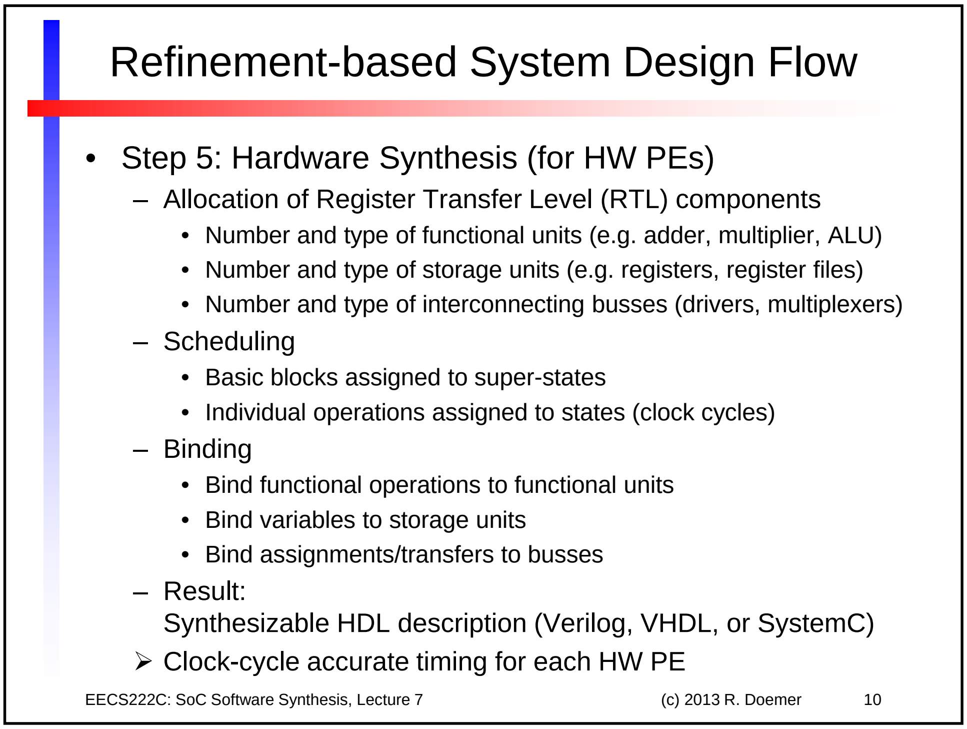
## Refinement-based System Design Flow

- Step 3: Network Refinement
  - Allocation of system busses
    - Number and type of system busses
    - Number and type of communication elements (CEs)
      - Transducers: Routers or bridges
    - System connectivity
      - Masters and slaves
  - Mapping of channels to busses and transducers
    - Map each inter-PE communication channel to a system bus (or multiple busses, if applicable)
    - Routing
  - Result:
    Network model of the system
  - ➢ Accurate representation of top-level system components and their connectivity



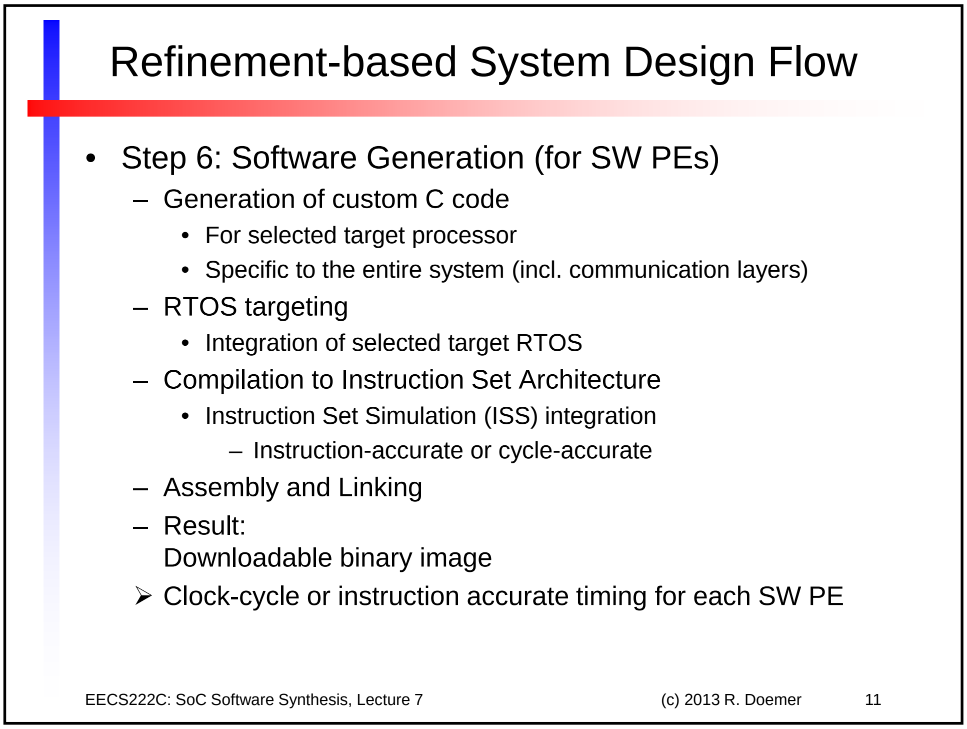
## Refinement-based System Design Flow

- Step 4: Communication Refinement
  - Allocation and specification of communication protocol(s) for each communication link (bus)
    - Type of bus protocol for each link (if applicable)
    - Bus protocol parameters (e.g. bit width, etc.)
    - Synchronization policy and parameters
      - Polling vs. interrupt
  - Mapping of addresses
    - System-wide address mapping to registers and memories
    - Address translation in transducers (if needed)
  - Result:
    Bus-functional model of the system
    - Transaction Level Model (TLM)
    - Pin Accurate Model (PAM)
  - ➢ Accurate timing for computation and communication



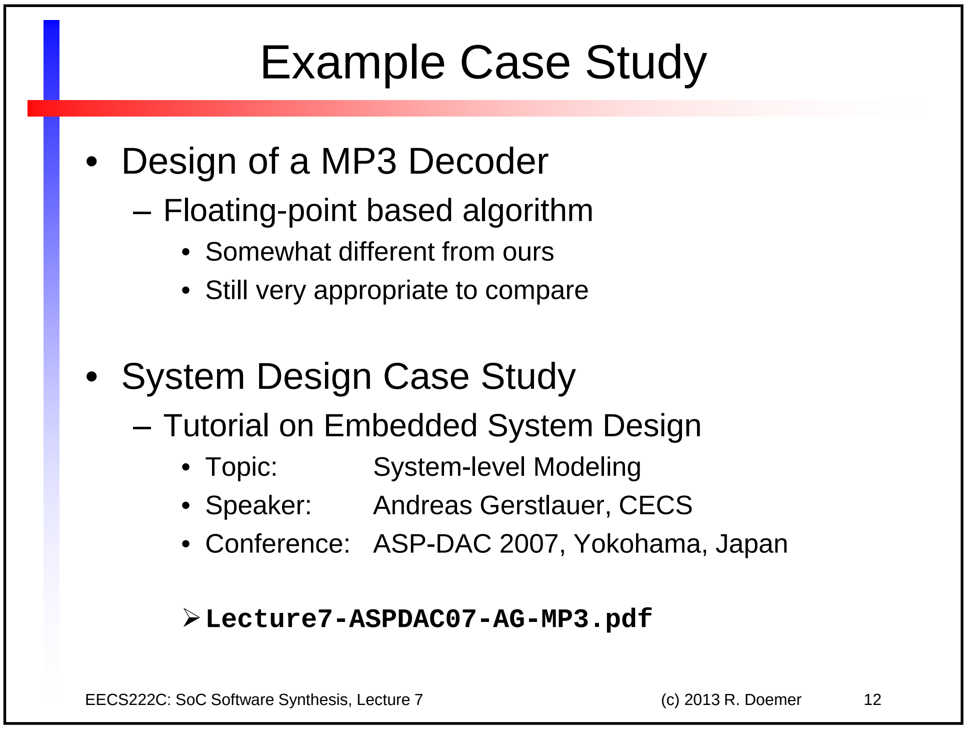
## Refinement-based System Design Flow

- Step 5: Hardware Synthesis (for HW PEs)
  - Allocation of Register Transfer Level (RTL) components
    - Number and type of functional units (e.g. adder, multiplier, ALU)
    - Number and type of storage units (e.g. registers, register files)
    - Number and type of interconnecting busses (drivers, multiplexers)
  - Scheduling
    - Basic blocks assigned to super-states
    - Individual operations assigned to states (clock cycles)
  - Binding
    - Bind functional operations to functional units
    - Bind variables to storage units
    - Bind assignments/transfers to busses
  - Result:
    Synthesizable HDL description (Verilog, VHDL, or SystemC)
  - ➢ Clock-cycle accurate timing for each HW PE



## Refinement-based System Design Flow

- Step 6: Software Generation (for SW PEs)
  - – Generation of custom C code
    - For selected target processor
    - Specific to the entire system (incl. communication layers)
  - – RTOS targeting
    - Integration of selected target RTOS
  - – Compilation to Instruction Set Architecture
    - Instruction Set Simulation (ISS) integration
      - – Instruction-accurate or cycle-accurate
  - – Assembly and Linking
  - – Result:
    Downloadable binary image
  - ➢ Clock-cycle or instruction accurate timing for each SW PE

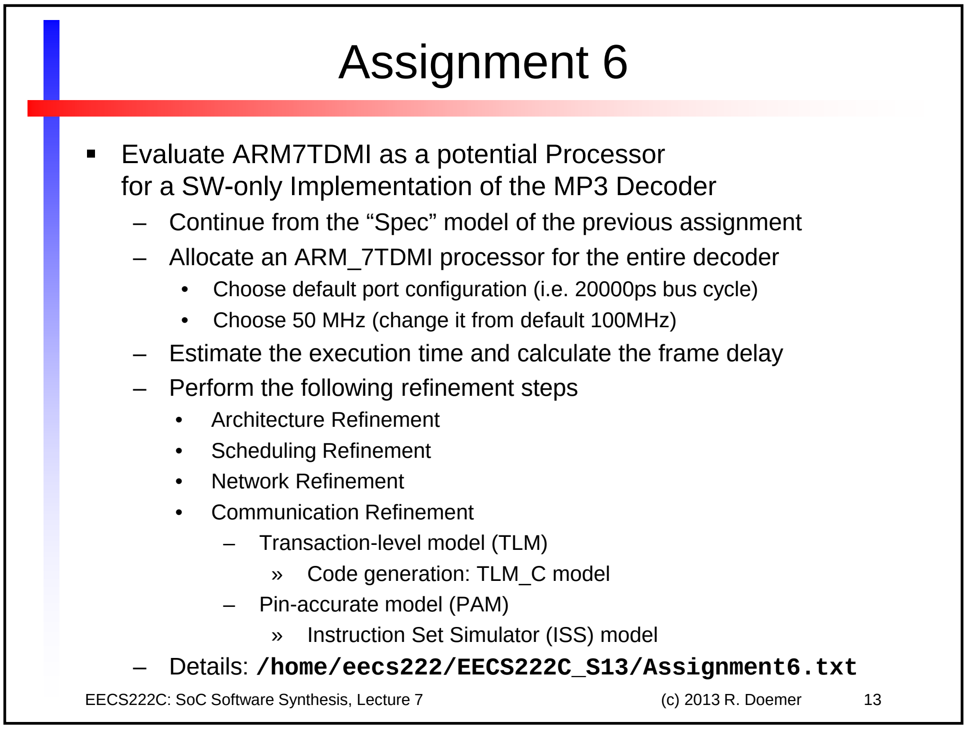


## Example Case Study

- Design of a MP3 Decoder
  - – Floating-point based algorithm
    - Somewhat different from ours
    - Still very appropriate to compare

- System Design Case Study
  - – Tutorial on Embedded System Design
    - Topic: System-level Modeling
    - Speaker: Andreas Gerstlauer, CECS
    - Conference: ASP-DAC 2007, Yokohama, Japan

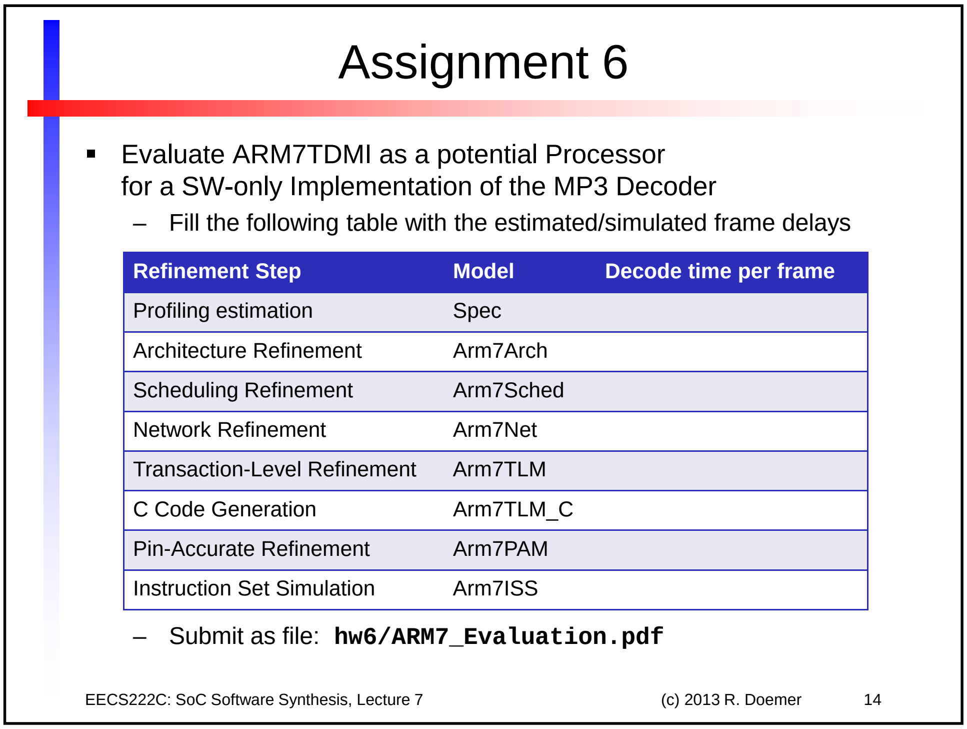
    ➢ **`Lecture7-ASPDAC07-AG-MP3.pdf`**



# Assignment 6

- Evaluate ARM7TDMI as a potential Processor
  for a SW-only Implementation of the MP3 Decoder
  - Continue from the “Spec” model of the previous assignment
  - Allocate an ARM_7TDMI processor for the entire decoder
    - Choose default port configuration (i.e. 20000ps bus cycle)
    - Choose 50 MHz (change it from default 100MHz)
  - Estimate the execution time and calculate the frame delay
  - Perform the following refinement steps
    - Architecture Refinement
    - Scheduling Refinement
    - Network Refinement
    - Communication Refinement
      - Transaction-level model (TLM)
        - » Code generation: TLM_C model
      - Pin-accurate model (PAM)
        - » Instruction Set Simulator (ISS) model
  - Details: **`/home/eecs222/EECS222C_S13/Assignment6.txt`**

EECS222C: SoC Software Synthesis, Lecture 7 (c) 2013 R. Doemer 13



# Assignment 6

- Evaluate ARM7TDMI as a potential Processor
  for a SW-only Implementation of the MP3 Decoder
  - Fill the following table with the estimated/simulated frame delays

| Refinement Step | Model | Decode time per frame |
|---|---|---|
| Profiling estimation | Spec | |
| Architecture Refinement | Arm7Arch | |
| Scheduling Refinement | Arm7Sched | |
| Network Refinement | Arm7Net | |
| Transaction-Level Refinement | Arm7TLM | |
| C Code Generation | Arm7TLM_C | |
| Pin-Accurate Refinement | Arm7PAM | |
| Instruction Set Simulation | Arm7ISS | |

  - Submit as file: **`hw6/ARM7_Evaluation.pdf`**

EECS222C: SoC Software Synthesis, Lecture 7 (c) 2013 R. Doemer 14